


JC715 U.S. PTO
09/28/00

JC827 U.S. PTO
09/28/00
09/28/00

UTILITY PATENT APPLICATION TRANSMITTAL <small>(Only for new nonprovisional applications under 37 CFR 1.53(b))</small>		Attorney Docket No. 197808US2RD
First Inventor or Application Identifier Masahiro ISHIYAMA		
Title	DOMAIN NAME SYSTEM INQUIRY APPARATUS, DOMAIN NAME SYSTEM INQUIRY METHOD, AND RECORDING MEDIUM	

APPLICATION ELEMENTS See MPEP chapter 600 concerning utility patent application contents		ADDRESS TO: Assistant Commissioner for Patents Box Patent Application Washington, DC 20231
1. <input checked="" type="checkbox"/> Fee Transmittal Form (e.g. PTO/SB/17) (Submit an original and a duplicate for fee processing) 2. <input checked="" type="checkbox"/> Specification Total Pages 52 3. <input checked="" type="checkbox"/> Drawing(s) (35 U.S.C. 113) Total Sheets 7 (Formals) 4. <input type="checkbox"/> Oath or Declaration Total Pages <input type="text"/> a. <input type="checkbox"/> Newly executed (original or copy) b. <input type="checkbox"/> Copy from a prior application (37 C.F.R. §1.63(d)) (for continuation/divisional with box 15 completed) i. <input type="checkbox"/> DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §1.63(d)(2) and 1.33(b). 5. <input type="checkbox"/> Incorporation By Reference (usable if box 4B is checked) The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4B, is considered to be part of the disclosure of the accompanying application and is hereby incorporated by reference therein.		ACCOMPANYING APPLICATION PARTS 6. <input type="checkbox"/> Assignment Papers (cover sheet & document(s)) 7. <input type="checkbox"/> 37 C.F.R. §3.73(b) Statement <input type="checkbox"/> Power of Attorney (when there is an assignee) 8. <input type="checkbox"/> English Translation Document (if applicable) 9. <input checked="" type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input checked="" type="checkbox"/> Copies of IDS Citations (2) 10. <input type="checkbox"/> Preliminary Amendment 11. <input checked="" type="checkbox"/> White Advance Serial No. Postcard 12. <input type="checkbox"/> Small Entity Statement filed in prior Certified Document(s) <input type="checkbox"/> application. Status still proper and desired. 13. <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) (1) (if foreign priority is claimed) 14. <input checked="" type="checkbox"/> Other: Notice of Priority, List of Inventor's Name and Address
15. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below: <input type="checkbox"/> Continuation <input type="checkbox"/> Divisional <input type="checkbox"/> Continuation-in-part (CIP) of prior application no.: Prior application information: Examiner: Group Art Unit:		
16. Amend the specification by inserting before the first line the sentence: <input type="checkbox"/> This application is a <input type="checkbox"/> Continuation <input type="checkbox"/> Division <input type="checkbox"/> Continuation-in-part (CIP) of application Serial No. Filed on <input type="checkbox"/> This application claims priority of provisional application Serial No. Filed		
17. CORRESPONDENCE ADDRESS  22850 (703) 413-3000 FACSIMILE: (703) 413-2220		

Name:	Marvin J. Spivak	Registration No.:	24,913
Signature:	<i>Chm McClelland</i>	Date:	9/28/00
Name:	C. Irvin McClelland	Registration No.:	

Registration Number 21,124

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

- INVENTOR(S) Masahiro ISHIYAMA

SERIAL NO: New Application

- FILING DATE: Herewith

FOR: DOMAIN NAME SYSTEM INQUIRY APPARATUS, DOMAIN NAME SYSTEM INQUIRY METHOD,
AND RECORDING MEDIUM

FEE TRANSMITTAL

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

FOR	NUMBER FILED	NUMBER EXTRA	RATE	CALCULATIONS
TOTAL CLAIMS	7 - 20 =	0	× \$18 =	\$0.00
INDEPENDENT CLAIMS	4 - 3 =	1	× \$78 =	\$78.00
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIMS (If applicable)			+ \$260 =	\$0.00
<input checked="" type="checkbox"/> LATE FILING OF DECLARATION			+ \$130 =	\$130.00
			BASIC FEE	\$690.00
TOTAL OF ABOVE CALCULATIONS				\$898.00
<input type="checkbox"/> REDUCTION BY 50% FOR FILING BY SMALL ENTITY				\$0.00
<input type="checkbox"/> FILING IN NON-ENGLISH LANGUAGE			+ \$130 =	\$0.00
<input type="checkbox"/> RECORDATION OF ASSIGNMENT			+ \$40 =	\$0.00
TOTAL				\$898.00

- ☐ Please charge Deposit Account No. 15-0030 in the amount of _____ A duplicate copy of this sheet is enclosed.
- ☒ A check in the amount of \$898.00 to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge any additional fees which may be required for the papers being filed herewith and for which no check is enclosed herewith, or credit any overpayment to Deposit Account No. 15-0030. A duplicate copy of this sheet is enclosed.

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124Date: 9/28/00

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 11/98)

DOMAIN NAME SYSTEM INQUIRY APPARATUS, DOMAIN NAME SYSTEM
INQUIRY METHOD, AND RECORDING MEDIUM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a domain name system (DNS) inquiry method capable of obtaining an appropriate response at high speed in a situation in which an inquiry must be made to a tree formed of a plurality of DNS servers due to relocation, disconnection of a line, etc., in a host which uses a DNS on the Internet.

2. Description of the Related Art

In recent years, use of the computer network, the "Internet", has spread worldwide. New computer businesses have been developed using information and services which are made publicly available as a result of connection with the Internet, and conversely, by providing information and services to outside users by access via the Internet. Also, regarding Internet uses, new technological developments and expansions have occurred.

On the Internet, each computer has an identifier called an "IP address", and packets are exchanged on the basis of these IP addresses. However, an IP address is a string of numbers, and it is difficult for human beings to remember such a string of numbers; nonetheless, in order to access

the computer through the Internet, the IP address must be specified. In order to solve this problem, a domain name system (DNS) has been devised. The DNS is a kind of database system, and one of the functions thereof is to mutually convert an IP address, which is a string of numbers and a fully qualified domain name (FQDN), which is easy for human beings to remember. That is, this is equivalent to an action of searching for an IP address corresponding to the FQDN in a database system for a DNS. An application which uses the Internet usually uses this DNS so as to convert a host name (this is usually an FQDN), which is input by a user, into an IP address, and to send data packets to the target site on the Internet.

The DNS is from one point of view a distributed database. It is impossible to manage all the hosts on the Internet by one database system. For this reason, the DNS takes the form of a distributed database having a tree structure. Each DNS server generally has the following three pieces of information.

- (1) Information about the server serving as a root
- (2) All the data of a specific domain
- (3) Information about the set of server to which a server itself has transferred authority, and transferred domain name

An FQDN is an example of a name space having a tree

structure which is divided by dots(.), for example, as in "foo.bar.com". This is a tree ".com bar foo" with the "." being the root. A leaf of a normal tree is a host name, and the other portions are called "domain mates". If a domain name "bar.com" is present on the Internet, a DNS server, which is informed of everything about the domain "bar.com", is present on the Internet.

Next, the manner in which the conversion between the domain name and the IP address is performed specifically in practice is described briefly. A DNS server which is located on the Internet in a distributed manner receives a request to convert "foo.bar.com" into an IP address from a computer which is a client. At this time, if the DNS server itself (let us assume that this is called "server A") receiving the request has all the data of "bar.com", the DNS server returns the converted result thereby. However, otherwise, the DNS server makes an inquiry to the root server. Since the root server does not have data for "bar.com", but is informed of the server which has transferred the authority of the "com" which is a branch of the tree, the root server informs the server of the server A. The server A makes an inquiry to the DNS server of "com". Since the DNS server "com" does not have the data of "bar.com", but is informed of the server which has transferred the authority of the "bar.com" which is a branch

of the tree, the DNS server informs the server of the server A. The server A makes an inquiry to the DNS server "bar.com". Since the server "bar.com" has all the data about "bar.com", it is possible to convert "foo.bar.com" into an IP address. This response is returned to the server A. The server A sends this response to the client.

In the manner as described above, since the DNS performs a search by searching the tree structure, even if an inquiry is made to any DNS server on the Internet, it is possible to always obtain the same response. Based on this premise, at the present time, a mobile computer is informed of a DNS server at the nearest location, and the DNS server is used.

However, this ensuring that "the same response can be obtained" is based on the precondition that the tree formed of DNS servers is unique on the Internet. However, in recent years, with the proliferation of firewalls, there are cases in which a DNS server tree which is independent within a firewall is constructed. Also, in SOHOs (Small Office, Home Office), etc., there are cases in which, connection to the Internet is intermittent, and in a network therewithin, there are cases in which an independent root server is provided.

In a case where a plurality of root servers are present in this manner, that is, a plurality of trees of DNS servers

are present, in the case of a mobile computer, even if a nearby DNS server is simply notified, the response varies depending on the tree to which that server belongs, presenting a problem. Specifically, if an FQDN is present in a certain tree and is inquired about a name server of a different tree, the response that will be returned is that such a host is not present.

There is also a system which makes inquiries to a plurality of DNS servers, and this method is a method in which inquiries are made in sequence until a server, from which a correct response is obtained, is encountered as a protocol of the DNS. In this method, since a correct response of "not found" can be obtained in a protocol of the DNS, there is a problem in that even if a plurality of DNS servers in respectively different trees are set, the response differs from the resulting determined response.

As described above, in the conventional method, in a case where there are a plurality of trees in which DNS servers are formed, the DNS server to which an inquiry should be made is unclear, and a desired response cannot be obtained.

SUMMARY OF THE INVENTION

The present invention has been achieved in consideration of the above-described circumstances. An

object of the present invention is to provide a DNS inquiry apparatus and a DNS inquiry method capable of obtaining a desired response at high speed even when a plurality of DNS trees are present.

To achieve the above-mentioned object, according to one aspect of the present invention, there is provided a domain name system inquiry apparatus comprising: current location information receiving means for receiving location information of the apparatus itself on a connected network; current location management means for storing location information received by the current location information receiving means; server information receiving means for receiving server information regarding a domain name system server to which an inquiry can be made; server management means for storing server information received by the server information receiving means; request receiving means for receiving an inquiry request to a domain name system server from a client; request transferring means for transferring the inquiry request received by the request receiving means to at least one domain name system server determined on the basis of the location information and/or the server information; response receiving means for receiving a response to the inquiry request transferred by the request transferring means; server information changing means for rewriting the server information when rewriting of the

server information occurs by the response received by the response receiving means; and request responding means for selecting a response result corresponding to the inquiry request on the basis of this server information and for sending the response result to the client.

As a result, there are no inquiries to DNS servers which are not most appropriate, making it possible to select a DNS server which is most appropriate for an inquiry.

Furthermore, the DNS inquiry apparatus of the present invention comprises algorithm receiving means for receiving an algorithm for selecting the response result; algorithm management means for storing the algorithm received by the algorithm receiving means; and algorithm processing section for selecting the response result in the request responding means by using the algorithm stored in the algorithm management means.

As a result, it is possible to change an algorithm when a selection is made.

Furthermore, according to another aspect of the present invention, there is provided a domain name system inquiry method comprising a first step of receiving the location information of an apparatus itself on a connected network; a second step of storing the location information received in the first step; a third step of receiving server information regarding a domain name system server to which an inquiry

can be made; a fourth step of storing the server information received in the third step; a fifth step of receiving an inquiry request to a domain name system server from a client; a sixth step of transferring the inquiry request received in the fifth step to at least one domain name system server determined on the basis of the location information and/or the server information; a seventh step of receiving a response to the inquiry request transferred in the sixth step; an eighth step of rewriting the server information when rewriting of the server information occurs by the response received in the seventh step; and a ninth step of selecting a response result to the inquiry request on the basis of this server information and sending the response result to the client.

In addition, the domain name system inquiry method of the present invention comprises a tenth step of receiving an algorithm for selecting the response result; an eleventh step of storing the algorithm received in the tenth step; and a twelfth step of selecting the response result in the ninth step by using the algorithm stored in the eleventh step.

Furthermore, according to yet another aspect of the present invention, there is provided a computer-readable recording medium having a domain name system inquiry method recorded therein, the domain name system inquiry method

comprising a first step of receiving the location information of an apparatus itself on a connected network; a second step of storing the location information received in the first step; a third step of receiving server information regarding a domain name system server to which an inquiry can be made; a fourth step of storing the server information received in the third step; a fifth step of receiving an inquiry request to a domain name system server from a client; a sixth step of transferring the inquiry request received in the fifth step to at least one domain name system server determined on the basis of the location information and/or the server information; a seventh step of receiving a response to the inquiry request transferred in the sixth step; an eighth step of rewriting the server information when rewriting of the server information occurs by the response received in the seventh step; and a ninth step of selecting a response result to the inquiry request on the basis of this server information and sending the response result to the client.

The above and further objects, aspects and novel features of the invention will become more fully apparent from the following detailed description when read in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a DNS inquiry apparatus according to an embodiment of the present invention;

Fig. 2 is an illustration of a current position table used in the embodiment of the present invention;

Fig. 3 is an illustration of a server management table used in the embodiment of the present invention;

Fig. 4 is an illustration of a request management table used in the embodiment of the present invention;

Fig. 5 is an illustration of a server information group table used in the embodiment of the present invention;

Fig. 6 is an illustration of a cache management table used in the embodiment of the present invention;

Fig. 7 is an illustration of a cache management table used in the embodiment of the present invention;

Fig. 8 is an illustration of a temporary inquiry processing table used in the embodiment of the present invention;

Fig. 9 is an illustration of a temporary inquiry processing table used in the embodiment of the present invention;

Fig. 10 is a configuration diagram of a computer network used in the embodiment of the present invention;

Fig. 11 is an illustration of a server additional-information table used in the embodiment of the present invention;

Fig. 12 is an illustration of an algorithm management table used in the embodiment of the present invention;

Fig. 13 is an illustration of a server additional-information table used in the embodiment of the present invention;

Fig. 14 is an illustration of a request management table used in the embodiment of the present invention;

Fig. 15 is a configuration diagram of a computer network used in the embodiment of the present invention; and

Fig. 16 is an illustration of a current position table used in the embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the present invention will now be described below with reference to the drawings.

Here, an inquiry origination source is called a "client", and a server in which a domain name system (DNS) is operating is called a "DNS server". The present invention relates to a method for relaying an inquiry to a DNS and receiving a response therefrom, and relates to an apparatus therefor.

An apparatus on the client side (mobile terminal) of this embodiment receives an inquiry issued from the client, and at the same time the apparatus makes an inquiry to a DNS server of which the apparatus on the client side is informed

at that time.

Here, an inquiry issued from the client is called an "original inquiry", and an inquiry which is newly issued from this apparatus is called a "temporary inquiry". A response from the DNS server is called a "temporary response", and a response to be sent to the client is called an "optimum response".

An inquiry to the DNS is identified by an ID contained in an inquiry packet. The client recognizes the fact that a response to its own inquiry has been returned by the fact that a response containing an ID equal to the ID inquired by the client itself has been returned. That is, in the inquiry, the set of the transmission source and the ID is unique. In this apparatus, the ID of the original inquiry is called an "original ID", and the ID used for a temporary inquiry is called a "temporary ID".

Fig. 1 shows a block diagram of this embodiment. This block diagram shows the construction of a DNS inquiry apparatus of the present invention, and it is assumed that the DNS inquiry apparatus is contained in a mobile terminal which is a client. Although a description is given here by using an example in which the DNS inquiry apparatus is contained in a mobile terminal, the construction may be formed in such a way that the DNS inquiry apparatus functions as an independent apparatus.

Referring to Fig. 1, a request receiving section 1 receives an original inquiry packet for a DNS server from the client. A request transferring section 2 transfers a temporary inquiry to the DNS server of which this apparatus is informed. A server information group management section 3 includes a server management section 4 and a current location management section 5, and manages information about DNS servers which are known at that time in a comprehensive manner. A cache management section 6 temporarily stores the result of the inquiry.

The current location management section 5 manages the location information of the network to which the mobile terminal is connected at that time. The server management section 4 manages the information about the DNS servers which are known at that time. A request management section 7 manages a temporary inquiry which is being issued at that time. A current location information receiving section 8 is a section for receiving the current location of a mobile host and for informing the current location management section 5 of that location.

A server information receiving section 9 receives information about DNS servers to which an inquiry can be made, and information therefor, and sends information about them to the server management section 4. An algorithm management section 10 manages an algorithm for selecting an

optimum response on the basis of the algorithm input from an algorithm receiving section 11.

An algorithm processing section 12 selects, from the obtained temporary response, an optimum response by using the algorithm. A request sending section 13 transmits the optimum response packet to the client. A response receiving section 14 receives a temporary response. A response selection section 15 processes the obtained temporary response.

The current location management section 5 is a section for managing the type of network address this apparatus and the mobile terminal, which is a client, have at that time. This information is called a "current location table". The current location table, as shown in Fig. 2, is composed of a current location address (A1NE1). Here, the number of pieces of information of the current location address may be two or more.

The server management section 4 has a server management table for managing the DNS servers which are known at that time, and a server additional information table for managing additional information for each server. The server management table, as shown in Fig. 3, is composed of the address of a DNS server, a failure counter, and the effective period of the server.

Here, the failure counter counts the number of times

failures in inquiries occurred continuously up to that time. The server additional information table is composed of a management number, a server location network area, a current location network area, an inquiry type, priority, and a cache capability flag, which are related to a DNS server.

Here, the management number is a natural number, which is unique in the table. The network area is composed of a network address and a network prefix length. If the network prefix length is 32 (in the case of IPv4), this indicates that the network address is equal to the address of the server, that is, this is additional information which is exclusively used for the DNS server.

Here, a tuple, which is additional information for the DNS server which is determined to be most appropriate, is selected.

The current location area is composed of a network address and a network prefix length. A tuple which matches the current location information is selected. In a case where a plurality of current locations are provided, a tuple in the high order of the table is selected.

The inquiry type indicates an inquiry type of a DNS. However, an "any" type which matches any inquiry type may be specified. The priority indicates the priority of the DNS server. The cache capability flag indicates whether a response can be transferred to the cache management section

6.

A request management section 12 has a request management table for managing information for inquiries such that temporary inquiries are being made. The request management table, as shown in Fig. 4, is composed of a temporary ID, an optimum response termination flag, all original inquiry packets, and a flag indicating a server information group table. The server information group table is formed of information obtained from the server information group management section 3, and is composed of a server address, a failure counter, priority, and a cache capability flag, which are shown in Fig. 5.

The request management section 12 adds an appropriate tuple to this request management table each time a new temporary inquiry is issued. Also, the request management section 12 deletes an appropriate tuple in accordance with a request from the response selection section each time processing for a certain original response is terminated.

The cache management section 6 is a section for storing the obtained temporary response and deleting it as required. The cache management section 6 has a cache management table. The cache management table, as shown in Figs. 6 and 7, is composed of a server address, all the original inquiry packets, and all the temporary response packets.

The response selection section 15 is a section for

processing the obtained temporary response. The response selection section 15 has a temporary response tentative storage table and a temporary inquiry processing table. The temporary response tentative storage table is a table for temporarily storing the obtained temporary response. The temporary response tentative storage table is composed of obtained temporary response packets.

The temporary inquiry processing table, as shown in Figs. 8 and 9, is composed of a temporary ID, a server information group table, and a temporary inquiry start time.

The algorithm management section 10 has an algorithm management table 12, which is an algorithm table, for managing algorithms for selecting an optimum temporary response. One algorithm for selecting the optimum temporary response is set for the inquiry type. The algorithm management table is composed of an inquiry type and an algorithm to be used. Here, in order to select the optimum temporary response, the algorithm can use the server information group table and the results of the temporary response, which have been obtained up to that time.

The operation of each section in Fig. 1 will now be described in detail.

The current location information receiving section 8 receives an address on the network to which a terminal in which this apparatus is mounted is connected at that time or

a lost address. The current location information receiving section 8 notifies this address to the current location management section 5.

The current location management section 5 receives information for the current location information from the current location information receiving section, and adds a tuple to the current location management table or deletes a tuple therefrom. Also, when a current location information request is received, a duplicate of the current location management table is output to the request source.

Here, the "location information" indicates the current location address in the network to which the mobile terminal is connected at that time. In this embodiment, an IP address which is given dynamically or statically on that network is used.

The request receiving section 1 receives an original inquiry packet from the client and transfers this packet to the request transferring section 2. The request transferring section 2 receives the original inquiry packet from the request receiving section 1 and obtains an inquiry type from this packet. This inquiry type is shown to the server information group management section 3, and a server information group table is obtained. This allows the request transferring section 2 to obtain, from the server information group table, information for a group of servers

to which inquiries are made.

Next, the request transferring section 2 creates one new temporary ID which is not equal to any temporary ID which is being used at that time. Then, a duplicate of the original inquiry data is created, and this temporary ID, the original inquiry data, a duplicate of the server information group table, and a creation request are transferred to the request management section 7. The request management section 7 informs the response selection section 15 of the temporary ID.

Next, the request management section 7 shows, regarding each server address within the server group table, the server address and the original inquiry data to the cache management section 6. When a temporary response packet can be obtained from the cache management section 6, a mark indicating that it is a cache is attached, the original ID within the packet is changed to a temporary ID, and this is sent to the response selection section 15.

When a temporary response packet cannot be obtained, a temporary inquiry which makes exactly the same inquiry as the original inquiry is created by using the temporary ID and this is transmitted to the server. That is, this means that, other than when the sending source for the temporary inquiry packet is a terminal in which this apparatus is operating, the other party to which an inquiry is made is a

DNS server, and the ID for inquiry is a temporary ID, a inquiry packet of a DNS does not differ from the original inquiry packet.

Next, when the temporary ID, the original inquiry data, the server information group table, and a creation request are received, the request management section 7 creates a new tuple in the request management table by using them. At this time, the optimum response termination flag is set to be "false".

Also, when the temporary ID and a deletion request are received, a corresponding tuple, that is, a tuple containing the received temporary ID, is deleted from the request management table. When the temporary ID and a confirmation request are received, the request management section 7 searches the request management table for a corresponding tuple, and transmits "true" to the section which has made the request when there is a corresponding tuple and transmits "false" when there is no corresponding tuple.

Also, when the temporary ID and an optimum response termination notification are received, the request management section 7 searches the request management table for a corresponding tuple, and when there is a corresponding tuple, the optimum response termination flag of that tuple is set to be "true".

Also, when the temporary ID and a request management

information request are received, the request management section 7 searches the request management table for a corresponding tuple, and returns the corresponding tuple. If there is no corresponding tuple, a null tuple is returned.

Next, when the inquiry type is received, first, the server information group management section 3 obtains the current location information from the current location management section 5. Then, a server management table is obtained from the server management section 4.

Next, the current location information and the inquiry type are shown to the server management section 4, and a sub-table of a server additional information table corresponding (that is, a set of tuples is formed in which the current location is contained most strictly in the current location area and the inquiry type is equal) thereto is obtained.

Regarding the sub-table of the server additional information table obtained here, a tuple corresponding (that is, the server address is contained most strictly in the server network area) to each server address within the server management table is obtained.

A tuple of the server group information table is created from this tuple and the tuple of the server management table, corresponding to the server address to which attention is directed at that time.

Regarding the tuple in which "any" is placed in the inquiry type, the tuple is changed to the inquiry type at that time. Eventually, a tuple of the server group information table is created for each server address within the server management table, and this becomes a server group information table.

Here, a tuple in which the priority is "0" is deleted. The obtained server group information table is output to the request source.

The server management section 4 adds or deletes an appropriate tuple of the server management table in accordance with an addition or deletion request from the server information receiving section 9. Also, the server management section 4 adds or deletes an appropriate tuple of the server additional information management table in accordance with an addition or deletion request for the server additional information from the server information receiving section 9.

When there is a request for the server management table, the server management section 4 makes a duplicate of the server management table and outputs it to the request source. Also, the server management section 4 checks the server management table periodically, and deletes a tuple whose effective time has expired at the time it is checked.

Next, when the server address and a failure-counter-

increase request for a server are received, the server management section 4 increments the failure counter of the specified server by one. Also, when the server address and a failure-counter-erasure request of a server are received, the server management section 4 sets the failure counter of the specified server to zero.

When there is a request for the current location information, the inquiry type, and the server additional information table, the server management section 4 obtains a duplicate of a tuple (in which the inquiry type and the current location information are contained in the current location area) for each tuple within the current location information table regarding each tuple within the server additional information table.

At this time, there must be one tuple for one server location network area. If there are a plurality of tuples in one server location network area, the inquiry type is checked first. If there is a tuple of an inquiry type equal to the inquiry type in which the inquiry type is shown to be "any type" in one server location network area, the tuples of "any type" are deleted.

In one server location network area, when there are a plurality of tuples in which the current location area contains one current location address within the current location information table, the tuple of the narrowest area

is made to remain, and the rest are deleted.

However, when there are a plurality of tuples in one server location network area, the tuple having the smallest management number area is made to remain, and the rest are deleted.

A sub-table of the server additional information table created thereby is transferred to the request section.

When the temporary response packet is received, the response receiving section 14 obtains a temporary ID of the temporary response from the received temporary response packet. The response receiving section 14 transmits the temporary ID and a confirmation request to the request management section 7. When the response is "true", the temporary response packet is transferred to the response selection section. If the response is "false", the temporary response packet is discarded.

When the temporary ID is received, the response selection section 15 transfers the temporary ID and a request management information request to the request management section, and request management information is obtained. A tuple of a temporary inquiry processing table is created based on the server information group table within the obtained request management information and the current time, and is added to the temporary inquiry processing table.

Also, the response selection section 15 checks if there is a tuple of the temporary inquiry processing table, in which a fixed period has elapsed. This period is programmable, and a user can specify it. If there is a tuple in which a fixed period has elapsed, a temporary ID is obtained first for each tuple. All the temporary responses which match the temporary ID are first taken out from the temporary response tentative storage table.

A server group table is obtained from the tuple in which a fixed period has elapsed. A server, which is in the server group table and for which a temporary response is not obtained, is checked, and for each of these servers, a failure-counter-increase request of the server is transmitted to the server management section 4.

Initially, all the temporary responses matching the temporary ID are deleted from the temporary response tentative storage table. The temporary ID and a request management information request are transferred to the request management section 7 so as to obtain request management information. An optimum request termination flag in the request management information is confirmed.

Here, if the optimum request termination flag is "false", a server failure response corresponding to the original inquiry packet is created, and this is transferred to the request sending section 13.

Then, the temporary ID and a deletion request are shown to the request management section 7, and a tuple of the target temporary inquiry processing table is deleted.

When the temporary response is received, the response selection section 15 obtains the temporary ID from the received response.

A tuple of the corresponding temporary inquiry processing table is obtained from the temporary ID. A request for a server management table is made to the server management section 4. The value of the failure counter for each server within the server information group table within the tuple of the temporary inquiry processing table is updated from the obtained server management table.

If the temporary response is not obtained from the cache, the content of the temporary response is not in error, and a cache capability flag matching the DNS server which has transmitted the temporary response within the server group table within the tuple of the temporary inquiry processing table is "true", a duplicate of the obtained temporary response is transferred to a cache management section. The obtained temporary response is added to the temporary response tentative storage table. The temporary ID and a request management information request are transferred to the request management section so as to obtain request management information. If the optimum

response termination flag of the request management information is "true", the processing is terminated. All the temporary responses matching the temporary ID are taken out from the temporary response tentative storage table by making a duplicate thereof.

The obtained temporary response set and the duplicate of the server information group table are transferred to the algorithm processing section, and waiting for a response is performed. If the response of the algorithm processing section is null, since this means that an optimum temporary response is not yet obtained at that time, this processing is terminated. If the response of the algorithm processing section is one element of the set of transferred temporary responses, this means that the obtained temporary response is an optimum temporary response.

Regarding the temporary ID which is being processed at that time, all the temporary responses equal to this temporary ID, which are in the temporary response tentative storage table, are deleted. The temporary ID and an optimum response termination notification are shown to the response management section. The temporary ID and a request management information request are shown to the request management section 7 in order to obtain request management information. An optimum response to be transmitted to the client is created from the obtained request management

information and the temporary ID. That is, of the optimum temporary response, the temporary ID is changed to the ID of the original inquiry packet so that the transmission source and the receiving destination are reversed compared to those of the original inquiry packet. Then, the created optimum response is transferred to the request sending section 13.

The request sending section 13 transmits the received optimum response to the client.

The algorithm management section 10 adds or deletes an appropriate tuple to the algorithm table in accordance with an algorithm addition or deletion request from the algorithm receiving section 11.

When the inquiry type and the algorithm request are received, the algorithm management section 10 transfers them to the section which has made a request for a tuple whose inquiry type is equal. If there is no tuple whose inquiry type is equal, a tuple of "any type" is transferred.

When a set of temporary responses is received, the algorithm processing section 12 checks the inquiry type, and shows the inquiry type and the algorithm request to the algorithm management section 10 in order to obtain an algorithm. By using the algorithm obtained here, it is determined whether the optimum temporary response is present within the set of received temporary responses. When the optimum temporary response is present within the set of

received temporary responses, the optimum temporary response is transferred to the section which has made the request, and the rest is discarded.

When the optimum temporary response is not present within the set of received temporary responses, a null is returned to the section which has made the request, and the received temporary responses are discarded.

The cache management section 6 checks a cache table periodically, and searches for cache data whose effective period has expired. The cache data whose effective period has expired is deleted from the cache table. Also, when the server address and the inquiry type are shown, the cache management section 6 searches the cache table for a temporary response in which the transmission party is equal to the server address and the inquiry type is equal. When there is a matching temporary response, a duplicate thereof is returned to the section which has made the request. If there is no matching temporary response, a null is returned to the section which has made the request.

Also, when a response packet is received, the cache management section 6 adds this response to the cache table.

Next, an example of the construction of a computer network in which the present invention is used is shown in Fig. 10.

A sub-network NE1 is a sub-network which belongs to an

open network (for example, the Internet) to which a plurality of organizations are connected.

Reference letter DE1 denotes a DNS server on an external network. A tree, in which DNS servers are formed, to which the external network belongs, is denoted as TE. A DNS server DE1 belongs to the DNS-structured tree TE.

In contrast, reference letter NA denotes a network (private.com) protected by a firewall server FW.

The firewall server FW is connected to the external network (for example, the Internet). Here, it is assumed that an access to the firewall server FW from the outside cannot be made.

The structured tree of the DNS servers DA1 and DA2 (not shown) inside this network NA is terminated in the network NA. This structured tree is denoted as TA. Reference letters NA1 and NA2 denote sub-networks belonging to the network NA. The DNS servers DA1 and DA2 (not shown) belong to the same structured tree TA.

Reference letter M1 denotes a mobile terminal which can be moved on the network. The mobile terminal M1 is a terminal having a DNS server inquiry function. It is assumed that the mobile terminal M1 is connected to the sub-network NE1 at that time, and that a dynamic or static address A1NE1 is obtained.

In this embodiment, it is assumed that "private.com" is

a domain name belonging to only the DNS-structured tree TA and is a domain name which can be used only in the network NA, and only a response of NXDomain (means "not known") can be obtained even if an inquiry is made to another DNS-server-structured tree.

In this embodiment, the inquiry packet of the DNS is abbreviated and denoted by the seven values of the transmission party, the destination server, the ID, the inquiry type, the inquiry data, and the response data. The table held by each section of this apparatus in this state is as described below.

The current position table is shown in Fig. 2. That is, the address indicating the current position of this terminal is A1NE1. The server management table is shown in Fig. 3. That is, this terminal knows DA1 and DA2 as the DNS servers to which inquiries can be made. Also, the DNS server DA1 has failed in inquiries five times continuously up to that time.

The effective period of the server DA1 in this terminal is a time ET1. When the server information is not updated by this date, the tuple of this server is deleted from the server management table.

The server additional-information table is shown in Fig. 11. The server additional-information table is composed of the network area where the DNS server is positioned, the

network area of the current position of this terminal, the inquiry type of each DNS server, the priority, and the cache capability flag.

The request management table is null at that time. That is, this means that any processing is not being performed at this time. In a similar manner, the cache management table, the temporary response tentative storage table, and the temporary inquiry processing table are set to be null.

The algorithm management table is shown in Fig. 12. This shows that a specified algorithm X is used for any inquiry type.

Here, as an example, the algorithm X is defined as follows. A temporary response is defined as a "good temporary response", in which the content of the temporary response is not NXDomain (means that it is correct as a response, but the server does not know the answer of the question), AnswerRR is not 0 (an effective answer is not contained in the response, and only reference information is contained), and the temporary response is not in error. If the temporary response is a good temporary response among the obtained temporary response, and a server having a priority higher than that of the server which responded with the good temporary response is not in the server management table or the temporary response of that server having the

higher priority is not a good temporary response, the temporary response from that server is most appropriate. Also, if every obtained temporary responses is not "good temporary response", and if all the temporary responses of all the servers in which the failure counter is equal to or smaller than a set value are held, the optimum response becomes that temporary response if NXDomain is present in the obtained temporary response. Also, if there is a temporary response, in which AnswerRR is 0, among the obtained temporary responses, the optimum response becomes that temporary response, and if all the obtained temporary responses are in error, the error becomes the optimum temporary response. If all the temporary responses of all the servers in which the failure counter is 3 or less are not held, since the optimum response cannot be determined yet, a null is returned.

This algorithm can be appropriately changed to another algorithm received from the algorithm receiving section 11.

Here, a terminal M1 makes a DNS inquiry in order to know the address for the name of "foo.private.com". This inquiry packet is written as (M1, M1, 4321, A, foo.private.com, null).

This apparatus contained in the mobile terminal M1 receives this packet by the request receiving section 1. This packet is sent to the request transferring section 2.

The request transferring section 2 obtains an inquiry type "A" from this packet.

Next, this inquiry type "A" is shown to the server information group management section 3 so as to make a request for a server information group table.

The server information group management section 3 receives this inquiry type "A" and creates a server information group table. Initially, a request for current location information is made to the current location information receiving section 8.

The current location information receiving section 8 returns the current position from the current position table shown in Fig. 2. Then, the server information group management section 3 obtains a server management table shown in Fig. 3 from the server management section 4.

Next, the server information group management section 3 shows the current location information and the inquiry type "A" to the server management section 4 so as to make a request for a sub-set of the server additional-information table.

Receiving this request, the server management section 4 creates a sub-set of the current server additional-information table shown in Fig. 11. This result is shown in Fig. 13, and this is returned to the server information group management section 3.

Next, the server information group management section 3 creates a server information group management table from the obtained server management table (Fig. 3) and the sub-set of the server additional-information table (Fig. 13). The obtained server information group management table is shown in Fig. 5, and this is returned to the request transferring section 2.

The request transferring section 2 creates a temporary ID. Here, it is assumed that "1" is assigned as the temporary ID. The request transferring section 2 transfers the created temporary ID "1",) the original inquiry (M1, M1, 4321, A, foo.private.com, null, and the obtained server group management table (Fig. 5) to the request management section 7 so as to create a request management table.

Receiving this creation request, the management section 7 adds a new tuple to the request management table.

As a result, the request management table becomes as shown in Fig. 14.

The request transferring section 2 transmits the created temporary ID "1" to the response selection section 15. The response selection section 15 shows the temporary ID "1" to the request management section 7 so as to make a request for request management information. Receiving this request, the request management section 7 returns a tuple having the temporary ID "1" within the request management

table.

More specifically, this is a line of the temporary ID "1" in Fig. 14. Receiving this tuple, the response selection section 15 adds a new tuple to the temporary inquiry processing table.

When the current time is assumed to be "CT1", the temporary inquiry processing table becomes as shown in Fig. 8. The request transferring section 2 makes an inquiry to the cache management section 6 about whether or not there is a response to the original inquiry (M1, M1, 4321, A, foo.private.com, null) with regard to each server, that is, DA1 and DE1, on the basis of the obtained server group management table (Fig. 5).

Since the cache table is null, the cache management section 6 returns a null to inquiries for both. Since the response from the cache management section 6 is a null, the request transferring section 2 creates temporary inquiries with regard to DA1 and DE1. There are two temporary inquiries which are created: (A1NE1, DA1, 1, A, foo.private.com, null) and (A1NE1, DE1, 1, A, foo.private.com, null).

At that time, since the terminal M1 is outside the protection area of the firewall server FW, and arrival at the DNS server DA1 is impossible, a response from the DNS server DA1 is not received.

The packet can reach the DNS server DE1 from the terminal M1. However, since "private.com" belongs to only the DNS-structured tree TA, the DNS server DE1 belonging to the DNS-structured tree TA cannot obtain an answer for "private.com". Therefore, the response of the DNS server DE1 becomes (DE1, A1NE1, 1, A, foo.private.com, NXDomain).

When the response of the DNS server DE1 reaches the terminal M1, the response receiving section 14 receives this response. The response receiving section 14 checks the temporary ID from the received packet, and makes an inquiry to the request management section 7 about whether or not the obtained temporary ID "1" is present in the request management table.

Since the temporary ID "1" is present in the request management table, this temporary response is transferred to the response selection section 15. The server management section 4 outputs the address of the DNS server DE1 and a failure-counter-erasure request of this server, and sets the failure counter of the DNS server DE1 to zero.

The response selection section 15 receives the temporary response and obtains the temporary ID. A tuple corresponding to the temporary ID is obtained from the temporary inquiry processing table (Fig. 8). Also, the server management table is obtained, and the failure counter is updated. Here, since the cache capability flag with

regard to the DNS server DE1 within the server group management table within the temporary inquiry processing table is "true", a request is made to the cache management section 6 so that this temporary response is cached. As a result, the cache table becomes as shown in Fig. 6.

Next, a request for a tuple of the request management table for the temporary ID "1" is made to the request management section 7. Here, since the optimum response termination flag is "false", the response selection section 15 adds this temporary response to the tentative storage table.

Next, a temporary response for the temporary ID "1" is duplicated from the temporary response tentative storage table. As a result, (DE1, A1NE1, 1, A, foo.private.com, NXDomain) is obtained. A set of these temporary responses and the server information group management table (Fig. 5) are transferred to the algorithm processing section 12.

The algorithm processing section 12 obtains the inquiry type "A", shows this to the algorithm management section 10, and obtains an algorithm. Although the temporary responses from all the servers are not received, since the failure counter of the DNS server DA1 is 5, which exceeds 3 defined by the algorithm, (DE1, A1NE1, 1, A, foo.private.com, NXDomain) becomes the optimum response.

Therefore, here, (DE1, A1NE1, 1, A, foo.private.com,

NXDomain) is returned to the response selection section 15. Since the temporary optimum response is obtained, the response selection section 15 first deletes the temporary response matching the temporary ID "1" from the temporary response tentative storage table.

As a result, the temporary response tentative storage table becomes null. Next, the request management section 7 is informed of the optimum response termination of the tuple of the request management table for the temporary ID "1". This causes the optimum response termination flag of the tuple of the request management table for the temporary ID "1" becomes "true".

Next, the temporary optimum response is converted into an optimum response. This causes the optimum response to become (M1, M1, 4321, A, foo.private.com, NXDomain).

This optimum response is transferred to the response selection section 15. Thereafter, it is assumed that the terminal M1 connects to the network NA1 by using a modem, etc., while holding the address A1NE1, and an address A1NA1 is obtained.

The network configuration diagram in this case is shown in Fig. 15. At this time, the current position table becomes as shown in Fig. 16.

It is assumed that, in this state, the terminal M1 makes a DNS inquiry to know the mail exchange address for

the name of "bar.private.com". This inquiry packet is assumed to be (M1, M1, 4322, A, bar.private.com, null).

A DNS server inquiry apparatus contained in the terminal M1 receives this packet by the request receiving section 1. This is transferred to the request transferring section 2.

The request transferring section 2 obtains the inquiry type "A" from this packet, and shows this inquiry type "A" to the server information group management section 3 so as to make a request for a server information group table. This result becomes as shown in Fig. 5 in a manner similar to the previous case. The request transferring section 2 creates a temporary ID. Here, it is assumed that "2" is assigned as a temporary ID. The request transferring section 2 transfers the created temporary ID "2", the original inquiry (M1, M1, 4322, A, bar.private.com, null), and the obtained server group management table (Fig. 5) to the request management section 7 so as to create a request management table.

Receiving this creation request, the request management section 7 adds a new tuple to the request management table. As a result, the request management table becomes as shown in Fig. 4. The request management section 7 transmits the created temporary ID "2" to the response selection section 15. The response selection section 15 adds a new tuple to

the temporary inquiry processing table.

Here, when the current time is assumed to be "CT2", the temporary inquiry processing table becomes as shown in Fig. 9.

The request transferring section 2 makes an inquiry to the cache management section 6 about whether or not there is a response for the original inquiry (M1, M1, 4322, A, bar.private.com, null) with regard to each DNS server, that is, DA1 and DE1, on the basis of the obtained server group management table (Fig. 5).

Since no cache matching this inquiry is present in the cache table, the cache management section 6 returns a null to inquiries for both. Since the response from the cache management section 6 is a null, the request transferring section 2 creates temporary inquiries with regard to the DNS servers DA1 and DE1. There are two temporary inquiries which are created: (A1NE1, DA1, 2, A, bar.private.com, null) and (A1NE1, DE1, 2, A, bar.private.com, null).

As a result, since it can be confirmed that the packet reaches the DNS server DA1 through the line of the modem, it is possible for the terminal M1 to receive a response from the DNS server DA1. Here, it is assumed that the response of the DNS server DA1 is (DA1, A1NA1, 2, A, bar.private.com, A3NA1) and that the response of the DNS server DE1 is (DE1, A1NE1, 2, A, bar.private.com, NXDomain).

Also, it is assumed that the packet has reached the terminal M1 earlier in the DNS server DA1. When the response of the DNS server DA1 reaches the terminal M1, the response receiving section 14 receives this response. Here, since the temporary ID "2" is present in the request management table, this temporary response is transferred to the response selection section 15.

The server management section 4 outputs the address of the DNS server DA1 and a failure-counter erasure request of this server, and sets the failure counter of the DNS server DA1 to zero.

The response selection section 15 receives the temporary response and obtains the temporary ID. Then, a tuple corresponding to the temporary ID is obtained from the temporary inquiry processing table (Fig. 8). Then, a server management table is obtained, and the failure counter is updated. As a result, the failure counter of the DNS server DA1 is set to 0.

Next, since the cache capability flag for the DNS server DA1 within the server group management table within the temporary inquiry processing table (Fig. 8) is "true", a request is made to the cache management section 6 so that this temporary response is cached. As a result, the cache management table becomes as shown in Fig. 7.

Next, a request for a tuple of the request management

table for the temporary ID "2" is made to the request management section 7. Here, since the optimum response termination flag is "false", the response selection section 15 adds this temporary response to the tentative storage table.

Next, a temporary response for the temporary ID "2" is duplicated from the temporary response tentative storage table. As a result, (DA1, A1NA1, 2, A, bar.private.com, A3NA1) is obtained.

The set of these temporary responses and the server information group management table (Fig. 5) are transferred to the algorithm processing section 12. The algorithm processing section 12 obtains the inquiry type "A", shows this to the algorithm management section 10, and obtains an algorithm. Although the temporary responses from all the DNS servers are not received, since this temporary response is a good temporary response and there is no DNS server having a higher priority, (DA1, A1NA1, 2, A, bar.private.com, A3NA1) is the temporary optimum response, and this is returned to the response selection section 15.

Since the temporary optimum response is obtained, the response selection section 15 deletes a temporary response matching the temporary ID "2" from the temporary response tentative storage table, and informs the request management section 7 of the optimum response termination of the tuple

of the request management table for the temporary ID "2". As a result, the optimum response termination flag of the tuple of the request management table for the temporary ID "2" becomes "true".

Next, this temporary optimum response is converted into an optimum response. The optimum response becomes (M1, M1, 4322, A, bar.private.com, A3NA1), and this is transferred to the request sending section 13.

As a result, even when the terminal M1 is connected to a network having a different DNS tree, it is possible to obtain an optimum response in each connection.

As has thus been described, according to the present invention, if connection to a DNS server is possible for an inquiry belonging to a tree formed of different DNS servers, it is possible to obtain a response immediately. Also, in a case where connection to a predetermined DNS server is impossible, a response from the DNS server after a set time is not expected. Therefore, even if connection to the DNS server is formed in any manner, it is possible for a user to obtain a more appropriate response at a high response speed in a range in which inquiries can be made.

Many different embodiments of the present invention may be constructed without departing from the spirit and scope of the present invention. It should be understood that the present invention is not limited to the specific embodiment

described in this specification. To the contrary, the present invention is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the invention as hereafter claimed. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications, equivalent structures and functions.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227

WHAT IS CLAIMED IS:

1. A domain name system inquiry apparatus comprising:
current location information receiving means for
receiving location information of the apparatus itself on a
connected network;

current location management means for storing location
information received by said current location information
receiving means;

server information receiving means for receiving server
information regarding a domain name system server to which
an inquiry can be made;

server management means for storing the server
information received by said server information receiving
means;

request receiving means for receiving an inquiry
request to a domain name system server from a client;

request transferring means for transferring the inquiry
request received by said request receiving means to at least
one domain name system server determined on the basis of
said location information and/or said server information;

response receiving means for receiving a response to
the inquiry request transferred by said request transferring
means;

server information changing means for rewriting said

server information when rewriting of said server information occurs by the response received by said response receiving means; and

request responding means for selecting a response result corresponding to said inquiry request on the basis of this server information and for sending the response result to said client.

2. A domain name system inquiry apparatus according to claim 1, further comprising:

algorithm receiving means for receiving an algorithm for selecting said response result;

algorithm management means for storing the algorithm received by said algorithm receiving means; and

algorithm processing section for selecting a response result in said request responding means by using the algorithm stored in said algorithm management means.

3. A domain name system inquiry method comprising:

a first step of receiving the location information of an apparatus itself on a connected network;

a second step of storing the location information received in said first step;

a third step of receiving server information regarding a domain name system server to which an inquiry can be made;

a fourth step of storing the server information received in said third step;

a fifth step of receiving an inquiry request to a domain name system server from a client;

a sixth step of transferring the inquiry request received in said fifth step to at least one domain name system server determined on the basis of said location information and/or said server information;

a seventh step of receiving a response to the inquiry request transferred in said sixth step;

an eighth step of rewriting said server information when rewriting of said server information occurs by the response received in said seventh step; and

a ninth step of selecting a response result to said inquiry request on the basis of this server information and sending the response result to said client.

4. A domain name system inquiry method according to claim 3, further comprising:

a tenth step of receiving an algorithm for selecting said response result;

an eleventh step of storing the algorithm received in said tenth step; and

a twelfth step of selecting the response result in said ninth step by using the algorithm stored in said eleventh

5. A computer-readable recording medium having a domain name system inquiry method recorded therein, the domain name system inquiry method comprising:

a second step of storing the location information received in said first step;

a fourth step of storing the server information received in said third step;

a sixth step of transferring the inquiry request received in said fifth step to at least one domain name system server determined on the basis of said location information and/or said server information;

an eighth step of rewriting said server information when rewriting of said server information occurs by the response received in said seventh step; and

a ninth step of selecting a response result to said

inquiry request on the basis of this server information and sending the response result to said client.

6. A domain name system inquiry apparatus comprising:
current location information receiving mechanism
configured to receive location information of the apparatus
itself on a connected network;

current location management mechanism configured to
store location information received by said current location
information receiving mechanism;

server information receiving mechanism configured to
receive server information regarding a domain name system
server to which an inquiry can be made;

server management mechanism configured to store the
server information received by said server information
receiving mechanism;

request receiving mechanism configured to receive an
inquiry request to a domain name system server from a
client;

request transferring mechanism configured to transfer
the inquiry request received by said request receiving
mechanism to at least one domain name system server
determined on the basis of said location information and/or
said server information;

response receiving mechanism configured to receive a

response to the inquiry request transferred by said request transferring mechanism;

server information changing mechanism configured to rewrite said server information when rewriting of said server information occurs by the response received by said response receiving mechanism; and

request responding mechanism configured to select a response result corresponding to said inquiry request on the basis of this server information and for sending the response result to said client.

7. A domain name system inquiry apparatus according to claim 5, further comprising:

algorithm receiving mechanism configured to receive an algorithm for selecting said response result;

algorithm management mechanism configured to store the algorithm received by said algorithm receiving mechanism; and

algorithm processing section mechanism configured to select a response result in said request responding mechanism by using the algorithm stored in said algorithm management mechanism.

ABSTRACT OF THE DISCLOSURE

A domain name system inquiry (DNS) apparatus and a domain name system inquiry method which are capable of obtaining a desired response at high speed even when there are a plurality of trees of domain name systems are provided. For this purpose, location information received by a current location information receiving device for receiving location information of the apparatus itself on a connected network is stored. Then, server information received by a server information receiving device for receiving server information about a DNS server to which an inquiry can be made is stored. Then, an inquiry request is transferred to at least one DNS server determined based on the location information and the server information, and based on a response therefor, the server information is rewritten, and a response result corresponding to the inquiry request is selected.

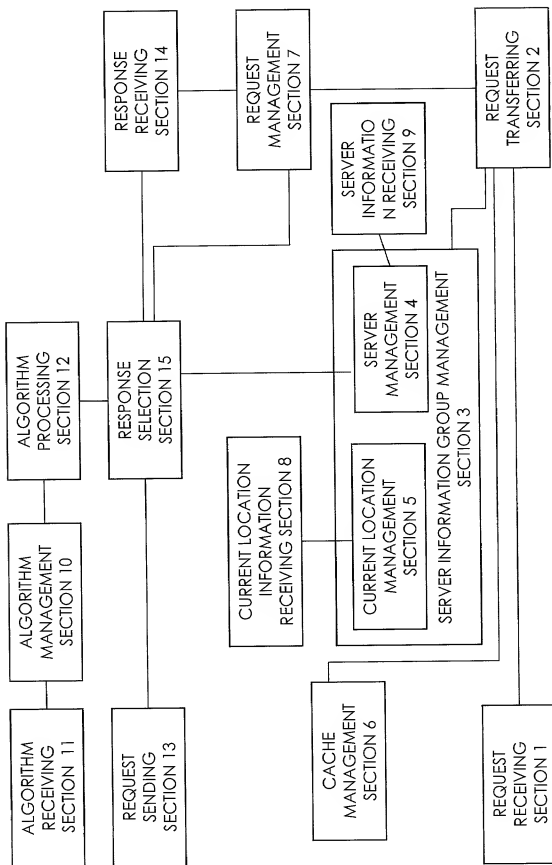


Fig.1

NETWORK ADDRESS AT PRESENT LOCATION
A1NE1

Fig.2

SERVER ADDRESS	FAILUE CONTER	EFFECTIVE PERIOD OF SERVER
DA1	5	ET1
DE1	0	ET2

Fig.3

SERVER ADDRESS	FAILUE CONTER	PRIORITY	CACHE CAPABILTY FLAG
DA1	5	1	TRUE
DE1	0	1	TRUE

Fig.5

CACHED TEMPORARY RESPONSE
(DE1,A1NE1,1,A.FOO.PRIVATE.COM,NXDomain)

Fig.6

TEMPORARY ID	OPTIMUM RESPONSE TERMINATION FLAG	ALL ORIGINAL INQUIRY PACKETS	SERVER INFORMATION GROUP TABLE
1	TRUE	(M1.M1.4321.A.FOO.PRIVATE.COM.NULL)	Fig.5
1	FALSE	(M1.M1.4322.A.FOO.PRIVATE.COM.NULL)	Fig.5

Fig.4

TEMPORARY ID	OPTIMUM RESPONSE TERMINATION FLAG	ALL ORIGINAL INQUIRY PACKETS	SERVER INFORMATION GROUP TABLE
1	FALSE	(M1.M1.4321.A.FOO.PRIVATE.COM.NULL)	Fig.5

Fig.14

CACHED TEMPORARY RESPONSE
(DE1,A1NE1,1,A,FOO.PRIVATE.COM,NXDomain)
(DA1,A1NE1,2,A,BAR.PRIVATE.COM,A3NA1)

Fig.7

TEMPORARY ID	SERVER INFORMATION GROUP TABLE	TEMPORARY INQUIRY START TIME
1	Fig.5	CT1

Fig.8

TEMPORARY ID	SERVER INFORMATION GROUP TABLE	TEMPORARY INQUIRY START TIME
1	Fig.5	CT1
2	Fig.5	CT2

Fig.9

INQUIRY TYPE	ALGORITHM
ANY	ALGORITHM X IS USED

Fig.12

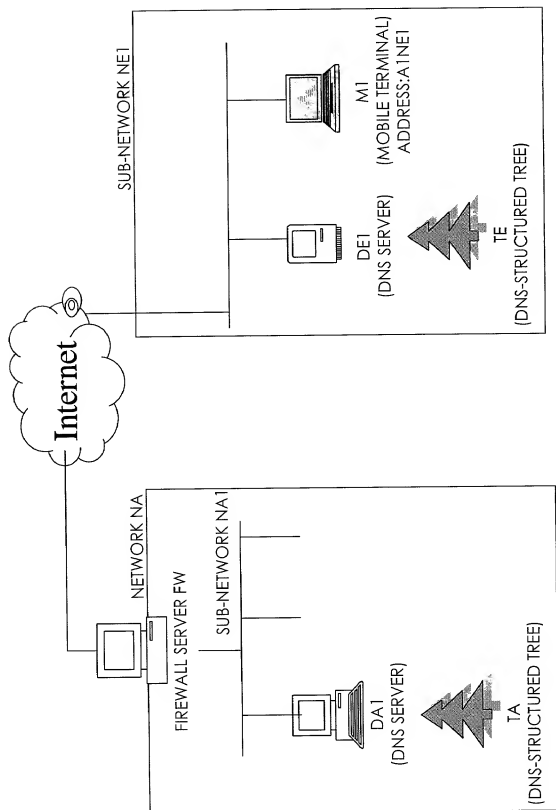


Fig. 10

MANAGEMENT ID	SERVER LOCATION NETWORK AREA	CURRENT LOCATION NETWORK AREA	INQUIRY TYPE	PRIORITY	CACHE VISIBLE FLAG
1	NA	ANY POSITION	ANY	1	TRUE
2	NA	ANY POSITION	MX	1	FALSE
3	NE	ANY POSITION	ANY	1	TRUE
4	ANY POSITION	ANY POSITION	ANY	2	TRUE

Fig.11

MANAGEMENT ID	SERVER LOCATION NETWORK AREA	CURRENT LOCATION NETWORK AREA	INQUIRY TYPE	PRIORITY	CACHE VISIBLE FLAG
1	NA	ANY POSITION	ANY	1	TRUE
2	NA	ANY POSITION	ANY	1	TRUE
3	ANY POSITION	ANY POSITION	ANY	2	TRUE

Fig.13

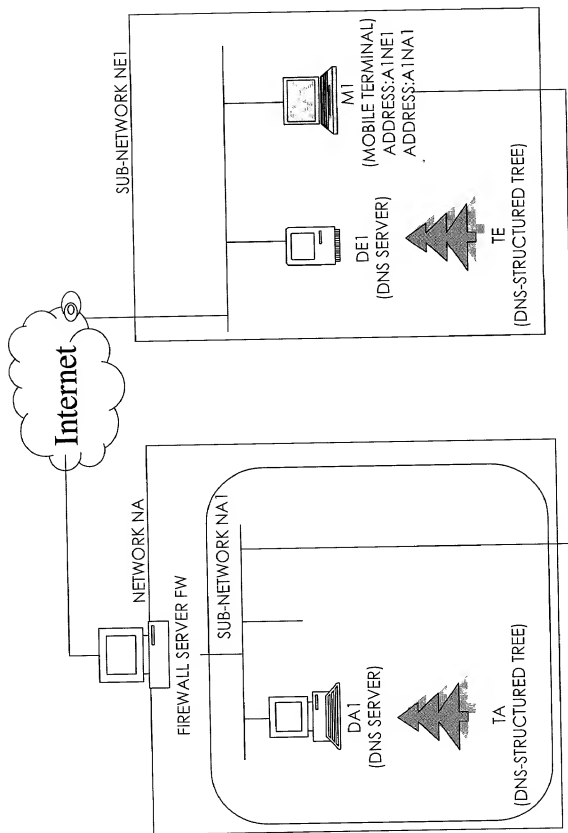


Fig.15

Docket No. 197808US2RD

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Masahiro ISHIYAMA

FILING DATE: Herewith

FOR: DOMAIN NAME SYSTEM INQUIRY APPARATUS, DOMAIN NAME SYSTEM INQUIRY METHOD,
AND RECORDING MEDIUM

1-c927 U.S. PTO
09/671245
09/28/00

LIST OF INVENTOR' S NAME AND ADDRESS

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

Listed below is the name and address of the inventor for the above-identified patent application.

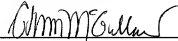
Masahiro ISHIYAMA

Tokyo, Japan

A declaration containing all the necessary information will be submitted at a later date.

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 11/98)